SARP Research Topics

Master List of Research Topics of Interest to the OSMA Software Assurance Research Program (SARP)

This list is not all inclusive

1. Tools and techniques for the assurance of systems that contain software. Specifically, assurance of the following types of software artifacts
   1.1. Requirements
   1.2. Specifications
   1.3. Design
   1.4. Code
   1.5. Test
   1.6. Interface
   1.7. Architecture

2. Methods and tools for detecting, identifying, and removing race conditions (via static and dynamic methods)

3. Tools to increase the efficiency of verification and validation activities, e.g.
   3.1. Traceability analyses
   3.2. Change propagation/impact analyses
   3.3. Re-verification analyses, i.e. tools that reduce the effort required to verify the resolution of identified defects, e.g. integration of CM/diff and defect tracking tools for all lifecycle product types.

4. Risk Assessment - Development of techniques for conducting and evaluating the correctness of probabilistic risk assessment (PRA) on software
   4.1. Use of artifact metrics to predict fault prone implementations
   4.2. Effectiveness of static versus dynamic metrics in predicting fault prone artifacts

5. Software Assurance practices for Auto-generated code
   5.1. Evaluation of available artifacts from Autocode Processes
   5.2. Verification of the Code Generator
   5.3. Automated analysis of the properties (e.g. constraint determination, temporal logic invariants, etc.) of available artifacts from the Autocode Process.

6. Software Assurance practices for COTS integration
   6.1. V&V of Interfaces to COTS
   6.2. Validation of a COTS application for an intended purpose

7. Use of simulators/testbeds in support of V&V
   7.1. Generic simulators/testbeds
   7.2. Reconfigurable simulators/testbeds
   7.3. Certification of simulators/testbeds
   7.4. Sensitivity analysis to determine required accuracy of the test environment

8. V&V of intelligent systems (autonomous and adaptive systems)
   8.1. Static analysis
   8.2. Model checking
   8.3. Runtime verification

9. Software assurance practices for reused/heritage software
   9.1. Reuse/heritage factors which impact software risk
   9.2. Appropriate level of software assurance for reused/heritage code. (criteria examples: mission similarities, flight hardware similarities, flight software similarities, ground system similarities)

9.3. Effective V&V techniques tailored to the common practice of reusing a spacecraft bus for new missions with less than the full set of lifecycle products

10. Reliability
    10.1. Reliability of OS
    10.2. Effects of changing operational profiles on software reliability (sensitivity)
    10.3. Impact of loading on software performance

11. Case Studies (What works and what doesn't)
    11.1. Characteristics of successful and unsuccessful software development projects
    11.2. Software development project risks resulting from incorrect cost estimating
    11.3. Software development project risk as a function of various management practices
    11.4. Reuse/product families
    11.5. Most costly software errors
    11.6. Effective methodologies (agile, OO, etc.)

12. Tandem experiments to improve software assurance (Excursions from current development, test, or V&V practices to determine the effectiveness of new practices.)

13. Transfer of best practices (Adaptation of best software development and testing practices to support some aspect of software assurance.)

14. IV&V
    14.1. Effectiveness of existing IV&V effort estimating tools (e.g. Risk Cube, CARA)
        14.1.1. Effectiveness of identification of error prone artifacts
        14.1.2. Effectiveness of analysis activities as applied to an artifact
        14.1.3. Effective tailoring of IV&V effort to desired risk reduction levels
    14.2. Practical model checking in support of IV&V
        14.2.1. Identification of appropriate techniques for IV&V model checking
        14.2.2. Identification of appropriate artifacts for IV&V model checking
        14.2.3. Identification of appropriate NASA projects for IV&V model checking
    14.3. IV&V of Software Development Processes
    14.4. Appropriateness of IV&V reused/heritage software (criteria examples: mission similarities, flight hardware similarities, flight software similarities, ground system similarities. (Note: this area is under-represented in the current funded research)
    14.5. Measures of IV&V effectiveness (e.g. Return on investments)

15. Benefits of software standards on the development of NASA software (e.g. FAA certification standards; e.g. RTCA DO-178B)

16. Assurance of field programmable gate arrays and Application-Specific Integrated Circuits (ASICs) (Note: this area is under-represented in the current funded research)